QPACA Evaluation

Submitted by Oregon Health & Science University
July 8, 2005

Description of this Document

The following document describes the evaluation of the QPACA web application made available to OHSU. It is divided into four sections:

1. General Description of QPACA
2. Initial Evaluation of QPACA
3. Usability issues with Current Version of QPACA
4. Preliminary Code Analysis
5. Appendix: A sample log.

Description of QPACA

QPACA (Quantitative Pathway Analysis in Cancer) is a Pathway Analysis system that consists of two components. The first is a visualization component for pathways that was not integrated with the version of QPACA that was evaluated and will not be discussed here. The second component allows researchers to input a list of genes that may be associated in a pathway and look for evidence of this association within a pathway within a microarray dataset.

Required Inputs

Two inputs are required:

1) A gene expression matrix, consisting of a large number (at least 50-60) of microarray samples that contains gene transcripts associated with the genes of interest.
2) A list of genes that may possibly be associated in a pathway. This list of genes is currently specified as an XML file where the type of ID and the ID name associated with the gene is identified to the program.

Output of QPACA

Currently, the web-enabled version of QPACA will return the best median correlation coefficient of all of those genes that are associated in the pathway to be studied. The optimized correlation coefficient of the pathway is compared with the correlation of both a set of randomized genes (simulating a pathway) and the pathway where the indvidual genes' expression profiles are scrambled (simulating the genes with similar expression values but a random expression profile).

Current Functionality of QPACA

At the time of evaluation, QPACA was integrated as a module in Magellan, a web-based analysis framework for data, which handles the data input functionality. The Magellan Module of QPACA runs the actual analysis and produced a number of files for further analysis. Perl Scripts were provided to parse the output files from the Magellan module, which will be integrated into the currently version of QPACA as time permits.

2. Initial Evaluation of QPACA

A special build of Magellan that integrated QPACA was made available to OHSU at http://cc-jainlab-sv1.ucsf.edu:8080/QPACA/. In addition, two large microarray datasets were made available to OHSU: the NCI60 Cancer dataset (60 samples) and the Hughes yeast compendium (300 samples).

Evaluation of QPACA was done using a set of already input pathway files made to OHSU by the UCSF developer. One pathway was chosen for evaluation: apoptosis. This pathway file was submitted to QPACA to examine the entire NCI60 Affymetrix Cancer dataset for evidence of these pathways. Parameters were submitted to QPACA: 600 optimization steps, 100 permutations of both random pathways and randomized gene profiles (see attached log file at the end of this document for additional parameters passed to QPACA).

When the optimization and randomization routines were completed, three output files representing the output were generated:

***pathwayname*.out** – The optimized output on the pathway itself
***pathwayname*.rgenes.out** – the output using random pathways (one null model), and
***pathwayname*.rdata.out** – the output using the same pathway, but scrambled genes.

These output files were parsed with Perl scripts in order to return the best median correlation values returned by the actual pathway analysis and the null models (both the false pathways and the scrambled gene data). The Perl script first finds the best median correlation for each output file, and then parses them into a single file called results.txt for comparison. These Perl scripts will be integrated into the current build of QPACA/Magellan to produce a more easy to understand output.

3. QPACA Usability Issues

General usability issues

1. *Parameter explanations*: If there were simple explanations of what each parameter did underneath each parameter, that would probably make the user feel a little less disoriented. One example would be for "randomize genes": simulate false pathways by selecting an unassociated group of genes in order to compare with real pathway.

2. *Nomenclature/Issues*: This ties into 1 above. The difference between optimization and permutation might be confusing to some users. Perhaps the permutation step can be more clearly associated with the null hypothesis in the user interface.

Issues with Web Interface

1. *Navigation Bar*: There is no easy way to get back from the main QPACA page labeled "Pathway Analysis" (http://cc-jainlab-sv1.ucsf.edu:8080/QPACA/AnalysisPages/Analyses/qpaca/selectPathwayParameters.jsp) to reselect Data. Some sort of simple navigation bar at the top of the page that would let you go back and forth, and reselect data would be helpful.

2. *File's Gene or LocusID type*: It took a while to figure out that the value of this field had to be the same as the id name of the pathway (such as 'locusid'). This might be a problematic part of the interface, so either there should be an explanation of where this value comes from, or perhaps this value could somehow be automated.

3. *Waiting for results*: Sometimes the Magellan session would expire while waiting for a result set. Bookmarking the intermediate analysis results page (http://cc-jainlab-sv1.ucsf.edu:8080/QPACA/AnalysisPages/waitForResults.jsp) was suggested, but if the user's session expires, there is no way to access the result set. If a URL could be provided to access the result set on this intermediate page (such as "results will be available at this URL"), or if you could have the program email the URL of the result set to the user when the analysis was finished, that would be helpful.

4. *Break up pathways parameters page into smaller pages*. I'm not sure if Magellan will allow you to do this, but the workflow of selecting samples, selecting annotation type, and then parameters is a bit confusing given the current layout. It might be less confusing if each of these had their own individual parameters pages.

Smaller Issues/Ideas

1. *Selecting samples can be a little unwieldy*. With 60 or more samples, if someone wanted to exclude only a few samples, this could unwieldy. If the "select all samples" check box would check all the boxes (maybe simple Javascript?) on the actual form, this could help with this problem.

2. *Use analysis log to resubmit a pathways analysis?* This might already part of your plan, but if someone wants to resubmit a pathways analysis, being able to upload a log file to recreate the analysis might be a nice feature.

Possible Bugs

Zero cutoff for missing values seemed to cause problems. I tried this on the NCI60 data set and entered a value of 0 for the missing values cutoff. My result set only consisted of the "others.txt" and "subset.txt" files and the analysis log. I tried this for various values of the other parameters and got similar results, even with different parameter files.

4. **Code Review for QPACA**

The QPACA algorithm (implemented in C) and the Perl scripts to parse the output files were made available to OHSU. The perl scripts parse the output file and are not described here.

The following is not a line by line code review, but pseudocode for readability.

Main Function

The main function in C controls the reading of the datafile and dataset to be studied and initializes the output matrices. It then performs the optimization step. If the randomized pathways option is selected, the program will select random genes from the dataset to simulate a pathway. If the randomized genes option is selected, the program will take the inputted pathway, shuffe the expression values around for each gene. The program then invokes the optimization step, and returns the largest median correlation value for each optimization step as output.

```
QPACA Main function(arguments)
        Parse arguments
        Initialize data array
                count rows and columns of datafile
                allocate memory for array
        Read datafile into program
        Initialize output matrices
        Initialize correlation matrices
        oloop = # of optimizations
        while(oloop is nonzero)
                if(randomizegenes is true)
                        pick a random bag of genes from dataset
                if(randomizedata is true)
                        shuffle expression values of selected genes in datamatrix
                iloop = # of random starts
                while(iloop is nonzero)
                        optimize(sizeofsubset, data, genes, correlationmatrix,
                                sizeofcormatrix)
                        output of optimize: .out file with sample sets picked and
                                best median correlation
                        iloop = iloop – 1
                end iloop
        oloop = oloop –1
        end oloop
end main
```

Optimize function

This function is the heart of the QPACA algorithm. It works by partitioning the dataset into two submatrices: the first is the subset to be tested, and the second consists of the remaining samples. These matrices are cloned into two test submatrices: the test subset, and the test remainder.

Optimize switches out two random samples between the test subset and the test remainder and runs the correlation on the test subset. If the median correlation is higher, then this switch is replicated in the subset and remainder set. If the median correlation is lower, then the samples are switched back in the test sets.

This is continued until the number of optimizations specified is achieved.

optimize(loop)
    find remaindersize: totalsize – subsetsize
    initialize subset matrix *subset*
    initialize remainder matrix *oset*
    initialize scratch subset matrix *tset*
    initialize scratch remainder matrix *toset*

    pick subset *subset* of size subsetsize (column indices on data)
    find remainder of set *oset* of size remaindersize
    copy *subset* to *tset*
    copy *oset* to *toset*

while(*loop* = nonzero)
    pick a random sample *i* from 1:*subsetsize*
    pick a random sample *j* from 1:*remaindersize*

    Switch the value at index *i* of *tset* with that of the value at index *j* of *toset*
        (swap one sample in *oset* for one in *tset*)
    Calculate correlation of tset using pearsons_correlation
        return median value *med*
    If (median value *med* is greater than previous best median value *tmed*)
        make *subset* equal to *tset*
        make *oset* equal to *toset*
        make *tmed* = *med*
        save current correlation matrix
    Else
        Swap values back such that
            tset = *subset*
            toset = *oset*
    loop = loop -1
    End loop
print out best median result

```
return best median
end optimize
```

5. Appendix: Sample log file from QPACA.

```
Data info:
      data set:
            Experiment number: 4
            Data type: Expression
            Annotation: Gene annotations for Staunton Affymetrix data
      gene or locus id in data file:      LocusLink
      samples chosen:
NSCLC_NCIH23,NSCLC_NCIH522,NSCLC_A549ATCC,NSCLC_EKVX,NSCLC_NCIH226,NSCL
C_NCIH332M,NSCLC_H460,NSCLC_HOP62,NSCLC_HOP92,COLON_HT29,COLON_HCC-
2998,COLON_HCT116,COLON_SW620,COLON_COLO205,COLON_HCT15,COLON_KM12,BREA
ST_MCF7,BREAST_MCF7ADRr,BREAST_MDAMB231,BREAST_HS578T,BREAST_MDAMB435,B
REAST_MDN,BREAST_BT549,BREAST_T47D,OVAR_OVCAR3,OVAR_OVCAR4,OVAR_OVCAR5,
OVAR_OVCAR8,OVAR_IGROV1,OVAR_SKOV3,LEUK_CCRFCEM,LEUK_K562,LEUK_MOLT4,LE
UK_HL60,LEUK_RPMI8266,LEUK_SR,RENAL_UO31,RENAL_SN12C,RENAL_A498,RENAL_C
AKI1,RENAL_RXF393,RENAL_7860,RENAL_ACHN,RENAL_TK10,MELAN_LOXIMVI,MELAN_
MALME3M,MELAN_SKMEL2,MELAN_SKMEL5,MELAN_SKMEL28,MELAN_M14,MELAN_UACC62,
MELAN_UACC257,PROSTATE_PC3,PROSTATE_DU145,CNS_SNB19,CNS_SNB75,CNS_U251,
CNS_SF268,CNS_SF295,CNS_SF539
Pathway description file: hsa04210.xml
      gene or locus id: locusid
Pathway analysis parameters:
      maximum number of optimizations:    600
      sample subset size: 20
      number of starting points: 20
Statistical permutation parameters:
      Randomize genes: Yes
      Randomize data: Yes
      Number random permutations:   500
Data modifications:
      Missing value cutoff: 0.0
      % allowed missing values: 0.3
      duplicate value method: var
      calculate ratios? Yes
      Log2 transformation? Yes
```